



ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΕΑΕΚ
ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ
ΣΥΓΧΡΗΜΑΤΟΔΟΤΗΣΗ
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
ΕΥΡΩΠΑΪΚΟ ΤΑΜΕΙΟ ΠΕΡΙΦΕΡΕΙΑΚΗΣ ΑΝΑΠΤΥΞΗΣ



ΠΑΙΔΕΙΑ ΜΠΡΟΣΤΑ
2^ο Επιχειρησιακό Πρόγραμμα
Εκπαίδευσης και Αρχικής
Επαγγελματικής Κατάρτισης

ΠΡΟΓΡΑΜΜΑ ΑΝΑΜΟΡΦΩΣΗΣ ΠΡΟΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΤΜΗΜΑΤΟΣ ΜΑΘΗΜΑΤΙΚΩΝ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

Μάθημα: Εισαγωγή στην Επιστήμη των Υπολογιστών

Θέμα: Σημειώσεις του Μαθήματος για το τμήμα (α):
(Από ΑΓΓΕΛΗ ΔΗΜΗΤΡΙΟ έως και ΔΟΥΡΑΜΑΝΗ ΔΗΜΗΤΡΙΟ)

ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΠΙΣΤΗΜΗ ΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

(Introduction to Computer Science)

Διδάσκων: Καθηγητής Κ.Ι. ΙΟΡΔΑΝΙΔΗΣ

Παραδόσεις: Τετάρτη, 11-13 (Υ35), Πέμπτη, 12-13 (062)

Γραφείο: Μ/Β 220 (Δέχεται καθημερινά - Πρωινά)

Φροντιστήρια-Εργαστήρια: Οι μεταπτυχιακοί φοιτητές:

Α.ΚΩΣΤΟΠΟΥΛΟΣ (Ο35) - Ι. ΝΙΚΑΣ (Ο15)

(α1): Τετάρτη 13-15 (Ο35 - Εργαστήριο Η.Υ.)

(α2): Τετάρτη 13-15 (Ο15 - Εργαστήριο Η.Υ.)

Κατά προτεραιότητα χρήση του εργαστηρίου:

(α2) Δευτέρα 17-19μμ. (α1) Τετάρτη 15-17μμ.,

καθώς και κάθε άλλη μέρα, όταν υπάρχουν θέσεις στο εργαστήριο.

ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΠΙΣΤΗΜΗ ΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ - ΤΜΗΜΑ (Α) ΑΚΑΔ. ΕΤΟΣ 2005 - 2006

(Ανάλυση Ύλης σε 13 Εκπαιδευτικές Εβδομάδες)

- 1η Εβδομάδα**
(Εργαστήριο 1ο και 2ο) Εισαγωγικά, Γενική Περιγραφή Συστατικών Στοιχείων ενός Υπολογιστή, Αριθμητική Υπολογιστών και Αριθμητικά Συστήματα, Συστήματα Μηχανικής Επεξεργασίας Πληροφοριών και Αλγόριθμοι Επίλυσης Προβλημάτων.
- 2η Εβδομάδα**
(Εργαστήριο 3ο και 4ο) Συνδυαστικά Κυκλώματα Πράξεων, Κυκλώματα Πολλαπλών Εισόδων, Κωδικοποιητές, Αποκωδικοποιητές, Συγκριτές Λέξεων, Πολυπλέκτες, Σχεδιασμός Μονάδων Αριθμητικής & Λογικής (ALU).
- 3η Εβδομάδα**
(Εργαστήριο 5ο και 6ο) Άλγεβρα Boole, Λογικές Συναρτήσεις, Απλοποιήσεις Λογικών Συναρτήσεων, Μέγιστοι και Ελάχιστοι Όροι, Χάρτης Karnaugh και Γραφική Παράσταση Λογικής Συνάρτησης, Εφαρμογές.
- 4η Εβδομάδα**
(Εργαστήριο 7ο και 8ο) Πίνακες Αληθείας, Θεώρημα Αναπαράστασης Λογικής Συνάρτησης, Βασική Τεχνική Υλοποίησης Λογικών Συναρτήσεων.

ΑΝΑΚΕΦΑΛΑΙΩΣΗ ΜΗΝΟΣ • 1ο Test Αφομοίωσης

ΒΑΣΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

1. «Ψηφιακή Σχεδίαση», του Morris M. Mano
2. «*Fundamentals of Computer Science*», του J.T. Colin
3. «Ηλεκτρονικοί Υπολογιστές», του Ν. Αλεξανδρίδη
4. «*FORTRAN 90 Programming*», του T.M.R. Ellis
5. «*Teach Yourself Java in 21 Days*», των L. LEMAY- C.L. PERKINS-
M.MORRISON

ΕΝΗΜΕΡΩΤΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

1. «*Η Επανάσταση του Κομπιούτερ*», του Chr. Evans (Εκδ. Γαλιλαίος)
2. «*Ο Ψύλος και οι γίγαντες*», του Eric Laurent (Εκδ. Ροές)
3. «*Megatrends*», του T. Maisbitt (Warner Books, 1984)

Ε.Ε.Υ. (1Η ΕΒΔΟΜΑΔΑ)

1η Εργαστηριακή άσκηση (Υπολογισμός εμβαδών πολυγώνων)

Δίδεται μια πολυγωνική περιοχή της μορφής του σχήματος 1. Οι πλευρές των 8 τριγώνων έχουν μετρηθεί καταλλήλως και ευρέθησαν ίσες με τις αναγραφόμενες τιμές.

Να γραφεί ένα πρόγραμμα που να υπολογίζει εμβαδά πολυγωνικών περιοχών (N γενικά τριγώνων) όταν δίδονται τα επί μέρους στοιχεία. Τέλος, το πρόγραμμα να εκτυπώνει ένα πίνακα της παρακάτω μορφής:

UNIVERSITY OF PATRAS-DEPARTMENT OF MATHS.

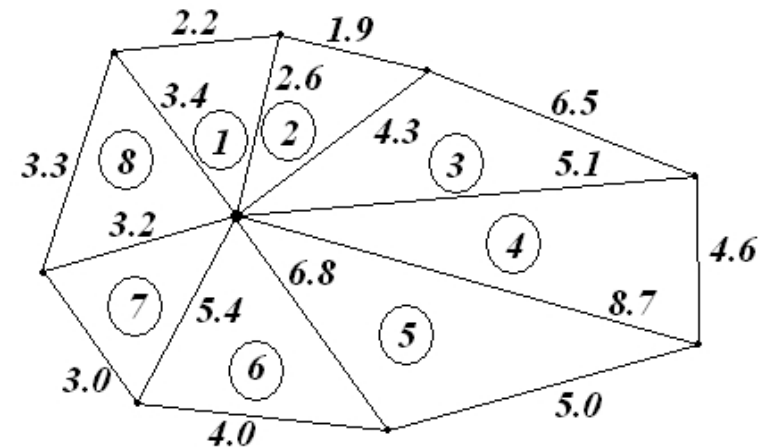
ANALYZING POLYGONAL DISTRICTS

+-+-+-

S/N	A	B	C	AREA
1				
2				
3				
4				
5				
6				
7				
8				

Total area:

Σχήμα 1.



2η Εργαστηριακή άσκηση (Εύρεση ριζών εξισώσεων)

Ένας τρόπος για να υπολογίσουμε μιά ρίζα μιας εξίσωσης:

$$\phi(x) = 0,$$

όταν η $\phi(x)$ είναι παραγωγίσιμη, είναι να εκτιμήσουμε με κάποιο τρόπο (π.χ. γραφικά) μια προσέγγιση της ρίζας x_0 , και στη συνέχεια να υπολογίσουμε ικανούς όρους της ακολουθίας με γενικό όρο:

$$(1) \quad x_{\nu+1} = x_{\nu} - \frac{\phi(x_{\nu})}{\phi'(x_{\nu})}, \quad \nu = 0, 1, 2, \dots$$

δηλαδή με βάση το x_0 βρίσκουμε τους επόμενους όρους της (1):

$$(2) \quad x_1 = x_0 - \frac{\phi(x_0)}{\phi'(x_0)}, \quad x_2 = x_1 - \frac{\phi(x_1)}{\phi'(x_1)}, \quad x_3 = x_2 - \frac{\phi(x_2)}{\phi'(x_2)}, \dots$$

Έτσι, εάν η παραπάνω ακολουθία συγκλίνει, τότε το όριο της x_{ν} είναι μια ρίζα της $\phi(x)$.

Σας δίδεται η εξίσωση:

$$(3) \quad xe^x - 1 = 0,$$

και μια εκτίμηση της ρίζας $x_0 \approx 0.5$ και σας ζητείται να γράψετε ένα πρόγραμμα FORTRAN που να υπολογίζει N (input) όρους της ακολουθίας με την βοήθεια των (2) (π.χ. $N = 20$), τους οποίους να εκτυπώνει με την παρακάτω επικεφαλίδα:

S/N	x^N
1	0.57102042
2	0.56715554
3	0.56714326
⋮	⋮
200	0.56714326

1η ΕΒΔΟΜΑΔΑ - ΔΕΥΤΕΡΑ 20 ΝΟΕΜΒΡΙΟΥ 2005

A. ΕΙΣΑΓΩΓΙΚΑ

1. Προλεγόμενα

Ο Ηλεκτρονικός Υπολογιστής (Η.Υ.) είναι μία μηχανή που είναι έτσι κατασκευασμένη ώστε (α) να προσλαμβάνει πληροφορίες, (β) να τις επεξεργάζεται κατάλληλα και (γ) να τις παρουσιάζει με τον ενδεδειγμένο τρόπο, που έχει ανάγκη ο χρήστης.

Το πρωταρχικό και κυριάρχο αγαθό στην όλη διαδικασία αξιοποίησης των Η.Υ. είναι η ΠΛΗΡΟΦΟΡΙΑ και κατά συνέπεια η ΓΝΩΣΗ.

Στην σημερινή κοινωνία, μια ευρεία κατηγορία απασχολήσεων συμπεριλαμβάνουν τη συλλογή, διανομή και παραγωγή πληροφοριών (συγγραφείς, δάσκαλοι, ταχυδρόμοι, δημοσιογράφοι, βιβλιοπώλες, κτλ.), όπου οι Η.Υ. ευρίσκονται στο άριστο πεδίο δράσης τους.

Εξ' άλλου πολλές φορές στον τύπο βλέπουμε την αναφορά ότι η κοινωνία μας βρίσκεται στο μέσο μιας «**επανάστασεως της πληροφορίας**». Στην κυριολεξία είναι **η τέταρτη κατά σειρά επανάσταση** της πληροφορίας στην πορεία της ανθρωπότητας! Η πρώτη αφορούσε **την ανάπτυξη του προφορικού λόγου**, η δεύτερη εμφανίστηκε με **την επινόηση του γραπτού λόγου** και τέλος **η τρίτη** αναπτύχθηκε με **την ανακάλυψη της τυπογραφίας**, στον Μεσαίωνα. Κάθε μία από τις παραπάνω επαναστάσεις, επέφερε βαθύτατη αλλαγή στην δομή της ανθρώπινης ζωής (με την πάροδο του χρόνου), φυσικά με ρυθμούς αντίστοιχους των ρυθμών της τότε ζωής.

Η επανάσταση της πληροφορίας της σημερινής εποχής οφείλεται στην ύπαρξη των ηλεκτρονικών υπολογιστών, οι οποίοι αν και αριθμούν μόνο ζωή μερικών δεκαετιών, έχουν τόσο βαθιά επηρεάσει τη ζωή μας, σε βαθμό αφάνταστο πριν την εμφάνισή τους.

Προφανώς, οι δυνατότητες που προσφέρουν οι σημερινοί Η.Υ. (**να μηχανοποιούν και πολλές διανοητικές εργασίες της καθημερινής ζωής παράλληλα με την τρομερή ταχύτητα επεξεργασίας στοιχείων που διαθέτουν και τις τεράστιες αποθηκευτικές ικανότητες που έχουν**), δίδουν το κλειδί της μεγάλης τους επιτυχίας. Εάν στα παραπάνω προστεθεί το μοναδικό γεγονός ότι μέχρι τα μέσα της δεκαετίας του '70 οι Η.Υ. ήταν πολυδάπανοι και έξω από τις οικονομικές δυνατότητες και τη ζωή του μέσου ανθρώπου, οι εξελίξεις όμως των τελευταίων ετών με τους μικροϋπολογιστές και τους «Οικιακούς Η.Υ.» έκαναν **εφικτή πραγματικότητα την είσοδο των Η.Υ. σε κάθε σπιτικό, λόγω του μικρού κόστους**. Η συνέπεια ήταν οι πρωτοφανείς δυνατότητες διευκόλυνσης της καθημερινής ζωής που παρουσιάστηκαν για άκοπη και αποτελεσματική επικοινωνία στις ανθρώπινες συναλλαγές και δραστηριότητες, με μια ασύλληπτη σε μέγεθος υποβοήθηση της ανθρώπινης κοινωνίας στη διανοητική επεξεργασία των διαφόρων προβλημάτων της.

Έτσι, σε μια κρίσιμη καμπή της κοινωνικής και οικονομικής ζωής της ανθρωπότητας, που η συνθετότητα των προβλημάτων που αντιμετώπιζε λόγω όγκου και πολυπλοκότητας, και που εφαινετο να την οδηγεί σε αδιέξοδο, σε αυτή τη δύσκολη περίσταση, η παρουσία των Η.Υ. έδωσε την υποδομή που αποτέλεσε το σχήμα της υπέρβασης και δημιούργησε την ελπίδα για ενδεδειγμένη αντιμετώπιση των προβλημάτων της ανθρώπινης κοινότητας, σε τέτοιο βαθμό που να θεωρείται σήμερα ότι **η πληροφορία και τα μέσα επεξεργασίας της**, που στην ουσία είναι γνώση και κατ' επέκταση Know How, **κατέχουν πλεονεκτικότερη θέση και από αυτή την άλλοτε «παντοδύναμη» ενέργεια**. Τέλος, η βέλτιστη αξιοποίηση των Η.Υ. - αυτού του «υπερόπλου» της μεταβιομηχανικής κοινωνίας που η διαμόρφωση του κορυφώνεται στις ημέρες μας με την προσπάθεια των Η.Υ. της «**πέμπτης γενιάς**» με την αυξημένη νοημοσύνη-προϋποθέτει ότι το κοινωνικό σύνολο θα έχει **ξεπεράσει την έννοια του υπολογιστή-μαγικό κουτί**. Σ' αυτή τη νέα πραγματικότητα αποσκοπούν να συμβάλλουν οι παραδόσεις αυτές που **μία φιλοδοξία έχουν, να είναι εύστοχες**.

2. Ανάγκες σημερινής πραγματικότητας

Οι ανάγκες της συγχρόνου εφαρμοσμένης επιστήμης και τεχνολογίας απαιτούν **ταχείες** και **ορθές** απαντήσεις σε καλώς καθορισμένες αριθμητικές διαδικασίες. Παράλληλα, οι ποσότητες πληροφοριών που δημιουργούνται, διαβιβάζονται και απαιτούν επεξεργασία στην καθημερινή μας ζωή, είναι τέτοιες που πολλές φορές μόνο **η άμεση και μαζική αντιμετώπιση των**, παρέχει τη δυνατότητα για **συναγωγή επωφελών και αξιοποιήσιμων συμπερασμάτων**.

3. Ο Η.Υ. ως όργανο θεραπείας

Ο ηλεκτρονικός υπολογιστής από καιρό ήδη έχει εδραιωθεί ως ένα **όργανο ανεκτιμήτου αξίας** για την αντιμετώπιση των ανωτέρω επιτακτικών αναγκών, με αφάνταστες προοπτικές για το μέλλον. Η χρήση του καθίσταται τόσο απαραίτητος στον σημερινό επιστήμονα, τεχνικό και επιχειρηματία ώστε η ίδια η ζωή να γίνεται αβίωτη χωρίς την παρουσία του.

Και τούτο, διότι οι εφαρμογές του σε επιστημονικά, τεχνικά και εμπορικά προβλήματα, καθώς και σε θέματα επεξεργασίας πληροφοριών, καθίσταται ολοένα και εκτενέστερες (ΔΕΗ, ΟΤΕ, Τράπεζες, Διοίκηση, κλπ.).

Άρα η γνώση μιας γλώσσας Η.Υ. αποτελεί **CONDITIO SINE QUA NON** («ὄρον, ἐκ τῶν ὧν οὐκ ἄνευ»).

4. Αιτιολόγηση

Βασικοί λόγοι που συνετέλεσαν στην εξέλιξη αυτή, είναι αφ' ενός μεν, τα τρία χαρακτηριστικά των υπολογιστών: **ταχύτητα, ακρίβεια και ικανότητα αποθήκευσης και επεξεργασίας** μεγάλων ποσοτήτων πληροφοριών· αφ' ετέρου δε, η **επαναληπτική** υφή πλείστων διαδικασιών της ανθρώπινης δραστηριότητας, όπου ο Η.Υ. ευρίσκεται στο άριστο δυνατό πεδίο δράσεώς του.

Τέλος, δεν πρέπει να παραγνωρίζεται και η μεγάλη ώθηση που επέδωσε **η τεράστια πρόοδος της Αριθμητικής Αναλύσεως** («αυτού του δεξιού χεριού των επιστημόνων...») στην ανάπτυξη **προσεγγιστικών μεθόδων**, με τις οποίες δύνανται να επιλυθούν άλυτα, η υπερμέτρως πολύπλοκα προβλήματα, κατά σχετικώς απλό και ικανοποιητικό τρόπο.

5. Φιλοσοφία των Η.Υ.

Τέλος, στο σημείο αυτό θα τονίσουμε πρώτον μεν, ότι, ο Η.Υ. **δεν επιλύει τα προβλήματα· απλώς εκτελεί τυφλώς και επακριβώς τις δοθείσες προς αυτόν εντολές** (σ' αυτή την τυφλή υπακοή του, οφείλεται άλλωστε και η προσωυμία του ως «**Ηλιθίου μεγάλης ταχύτητας**» που του προσεδόθηκε). Δεύτερο δε, να παρατηρήσουμε ότι ο Η.Υ., η κορωνίδα αυτή της συγχρόνου τεχνικής προόδου, **εκτελεί βασικώς μία και μόνο αριθμητική πράξη**, την πρόσθεση (ή, ισοδυνάμως, την αφαίρεση).

Όλα δε τα θαυμαστά που επιτελεί οικοδομούνται με την βοήθεια της Αριθμητικής Αναλύσεως και με την πρόσθεση σαν δομικό λίθο· π.χ., ο πολλαπλασιασμός εκτελείται με διαδοχικές προσθέσεις, η διαίρεση με διαδοχικές αφαιρέσεις, ενώ η αφαίρεση δύναται, με τη χρήση του συμπληρώματος, να αναχθεί σε πρόσθεση κ.τ.ο. Έτσι λοιπόν, από τη σκοπιά της «αριθμητικής», ο σύγχρονος Η.Υ. δεν διαφέρει ουσιαστικά από τις πρώτες υπολογιστικές μηχανές.

Σημείωση: Ως ένα παράδειγμα, ας πάρουμε την αφαίρεση δύο **διψηφίων** αριθμών των: 96-29, που γράφεται διαδοχικά:

$$96 - 29 = 96 - (100 - 71) = 96 + 71 - 100 = 167 - 100 = 67.$$

Παρατηρήσατε, ότι η τελική διαφορά υλοποιείται με διαγραφή του τρίτου ψηφίου που δημιουργήθηκε, **δηλαδή του 1**, πράγμα που γίνεται αυτόματα στους υπολογιστές ως το ψηφίο πλεονάσματος (overflow), που απορρίπτεται. Τέλος, ο αριθμός 71 αποτελεί **το συμπλήρωμα του 29**, ως προς τη βάση δεκαδικού αριθμητικού συστήματος, θέμα στο οποίο θα επανέλθουμε αργότερα.

6. Διάκριση των Η.Υ.

Εδώ, είναι σκόπιμο να διακρίνουμε τους Η.Υ. **σε τρεις μεγάλες κατηγορίες**: δηλαδή, στην κατηγορία των **ψηφιακών ή αριθμητικών*** Η.Υ. (Digital Computers), στην κατηγορία των **αναλογικών†** Η.Υ. (Analog Computers) και στην κατηγορία των **μεικτών Η.Υ.** (Hybrid Computers).

*Το βασικό χαρακτηριστικό τους είναι ότι **μετρούν· το αβάνκιο** είναι ένας ψηφιακός υπολογιστής.

†Το βασικό τους χαρακτηριστικό είναι **ότι συγκρίνουν με βάση την αναλογία**: ο λογαριθμικός κανόνας είναι ένας αναλογικός υπολογιστής.

Από τις προηγούμενες τρεις κατηγορίες Η.Υ. **η κατηγορία των ψηφιακών Η.Υ. είναι γενικής εφαρμογής**, αυτός δε είναι ο λόγος που είναι και η πλέον διαδεδομένη. **Όλες οι γλώσσες προγραμματισμού αναφέρονται στους ψηφιακούς Η.Υ.**, τους οποίους και μόνο θα έχουμε υπ' όψη στη συνέχεια.

7. Το γενεαλογικό δένδρο των Η.Υ.

Μια σύντομη ματιά στο «γενεαλογικό δένδρο» των σύγχρονων ηλεκτρονικών υπολογιστών μας φέρει στον 17ο αιώνα μ.Χ., όπου κάνει την εμφάνισή της **η μηχανή του B. Pascal (1624)**, που θεωρείται η πρώτη υπολογιστική μηχανή **με δυνατότητες εκτελέσως προσθέσεων και αφαιρέσεων**. Στη συνέχεια, ο **W. Leibnitz (1694)** έδωσε ένα βελτιωμένο τύπο μηχανής **που μπορούσε, επιπροσθέτως, να εκτελεί πολλαπλασιαμούς και διαιρέσεις**.

Μετά ακολούθησε μεταβατική περίοδος αιώνος, για να φθάσουμε στο 1812, όταν ο **C. Babbage**, αυτός ο Νεύτων των υπολογιστών, συνέλαβε την ιδέα της «**Μηχανής Διαφορών**» (Difference Engine) για την επίλυση διαφορικών εξισώσεων. Το 1833 η μεγαλοφυΐα του Babbage και η διάπυρος επιθυμία του για ακρίβεια, σχεδίασαν και ανέπτυξαν την «**Αναλυτική Μηχανή**» (Analytical Engine), που περιείχε όλα τα στοιχεία των σημερινών ηλεκτρονικών υπολογιστών. Δυστυχώς, η τεχνολογία της εποχής δεν συνεβάδιζε με την πρωτοπορεία των συλλήψεών του, όπως επίσης και οι άνθρωποι γύρω του (εκτός μερικών εξαιρέσεων, όπως της Λαίδης Lovelace-κόρης του Lord Byron) δεν κατάλαβαν τις ιδέες του με αποτέλεσμα να έπρεπε να παρέλθει ένα ακόμα αιώνας για την πραγμάτωση των ηλεκτρονικών υπολογιστών.

Εν τω μεταξύ, το 1886, ο **H. Hollerith** επινόησε μηχανικό σύστημα επεξεργασίας **διάτρητων δελτίων** για διαλογές και πινακοποιήσεις απογραφικών στοιχείων.

Τέλος, ο **H. Aiken**, του Πανεπιστημίου του Harvard, εργαζόμενος από το 1937 ολοκλήρωνε το 1944 τον **πρώτο πλήρως αυτοματοποιημένο ηλεκτρομηχανικό υπολογιστή, τον MARK I**, με τις εξής ταχύτητες εκτελέσεως των βασικών πράξεων: 0.33" για την πρόσθεση και αφαίρεση, 5" για τον πολλαπλασιασμό και έως 16" για τη διαίρεση. Παράλληλα, οι ανάγκες του πολέμου (βλέπετε, ο Ηράκλειτος είχε κάποια βάση, όταν έλεγε ο «ο πόλεμος πατήρ πάντων») και τα μέσα που διετέθησαν στους **J. Mauchly** και **J. Eckert**, του Πανεπιστημίου της Πενσυλβανίας, στην περίοδο 1942–1945, έκαναν πραγματικότητα τον **ENIAC** (Electronic Numerical Integrator and Calculator), **που αποτελεί τον πρώτο ηλεκτρονικό υπολογιστή**. Ο ENIAC περιείχε 18.000 ηλεκτρονικές λυχνίες, ζύγιζε 30 τόννους και κατελάμβανε 15.000 τετραγωνικά πόδια. πάντως ήταν 60 φορές ταχύτερος από ο,τιδήποτε άλλο σύστημα της εποχής του και εχρησιμοποιείτο κυρίως για τον υπολογισμό τροχιών βλημάτων.

8. Χαρακτηριστικά γενεών Η.Υ. και προοπτικές

Στα επόμενα 30 χρόνια αναπτύξεως των υπολογιστών, έλαβε χώρα μια πραγματικά επαναστατική πρόοδος.

Την πρώτη γενεά των ηλεκτρονικών υπολογιστών, **με έμβλημα την ηλεκτρονική λυχνία**, ακολούθησαν 3 άλλες γενεές, με χαρακτηριστικά τα **τρανζίστορς** (Transistors), **τα ολοκληρωμένα κυκλώματα** (Integrated circuits), και **την ολοκλήρωση μέσης έκτασεως** (MSI), ενώ η επικείμενη πέμπτη γενεά υπολογιστών θα ενσωματώνει την **ολοκλήρωση μεγάλης εκτάσεως** (LSI), **την ολοκλήρωση πολύ μεγάλης εκτάσεως** (VLSI) καθώς και **την ολοκλήρωση πάρα πολύ μεγάλης εκτάσεως** (ULSI). Στον πίνακα 1, για την εναργέστερη εικόνα της πραγματοποιούμενης εξελίξεως, παραθέτουμε μερικά βασικά χαρακτηριστικά των πέντε γενεών Η.Υ. (E. Joseph, Science, October 1970).

Πίνακας 1. Γενικά χαρακτηριστικά γενεών Η.Υ.

Γενεά	Ημερομηνία Εισαγωγής	Ρόλος	Λογικά Κυκλώματα	Ταχύτητα (sec)	Πλήθος Βασικών Εντολών
Πρώτη	1951–52	Εκτέλεση Υπολογισμών	Ηλεκτρονικές Λυχνίες	$10^{-2} - 10^{-4}$	< 100
Δευτέρα	1958–60	Επεξεργασία Δεδομένων	Τρανζίστορς	10^{-5}	περ. 100
Τρίτη	1963–65	Επεξεργασία Πληροφοριών	Ολοκληρωμένα Κυκλώματα	$< 10^{-6}$	περ. 200
Τετάρτη	1969–72	Πληροφορίες Αμέσου Προσπελάσεως	Ολοκλήρωση MSI/LSI	10^{-7}	> 200
Πέμπτη	1975–	Διανοητική Βοήθεια	Ολοκλήρωση VLSI/ULSI	$> 10^{-8}$	> 1000

Β. ΓΕΝΙΚΗ ΟΡΓΑΝΩΣΗ ΤΩΝ Η.Υ.

1. Ορισμός

Σαν Η.Υ. θα εννοούμε ένα αυτοματοποιημένο ηλεκτρονικό σύστημα επεξεργασίας αριθμητικών δεδομένων βάσει ενός προκαθορισμένου συνόλου εντολών (του προγράμματος) και σύμφωνα με τους νόμους της αριθμητικής και λογικής που είναι ενσωματωμένοι σ' αυτό. Τα δεδομένα αποτελούνται από αριθμούς και κωδικοποιημένες πληροφορίες που παρίστανται στη μνήμη του Η.Υ. με την **κατάσταση** ενός συνόλου από στοιχεία (τα bits-Binary Digits), κάθε ένα από τα οποία είναι δυνατό να ευρεθεί σε μία από δύο συγκεκριμένες καταστάσεις: **κάθε κατάσταση παριστά κι ένα ψηφίο, 0 ή 1**. Προφανώς, η **δυναδική αριθμητική** (Binary Arithmetic) αποτελεί τη «φυσική» αριθμητική του υπολογιστού, όπως ακριβώς η δεκαδική αριθμητική αποτελεί την ανθρώπινη φυσική αριθμητική.

2. Δομή των Η.Υ.

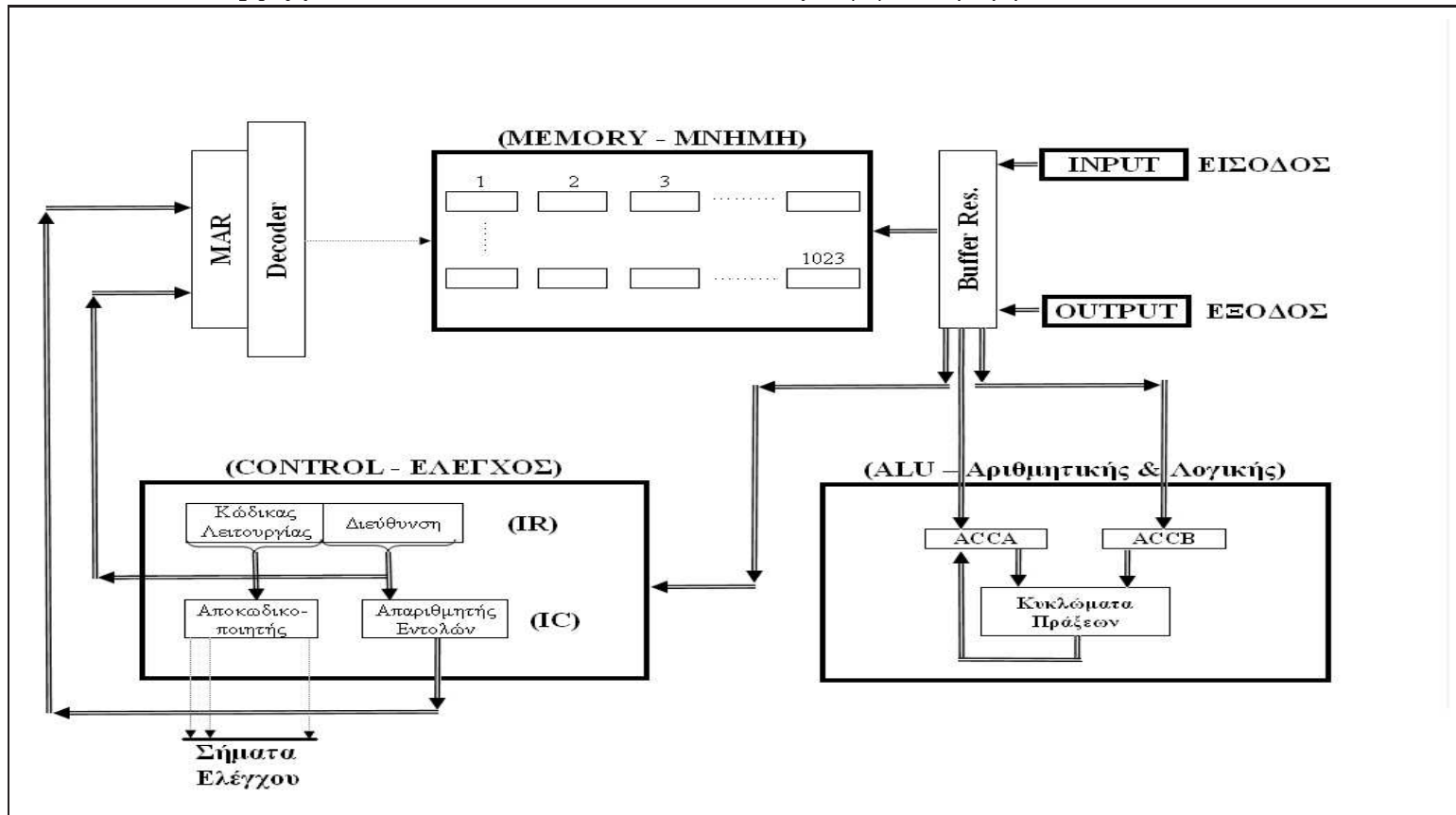
Η βασική δομή των Η.Υ. αποδίδεται εύγλωττα από το γνωστό **κλαστικό διάγραμμα ενός ψηφιακού Η.Υ.**, με τους αντίστοιχους καταγραφείς (registers), που παρατίθεται στο σχήμα 1.

Από το σχήμα 1 καθίστανται σαφή τα **5 βασικά τμήματα ενός Η.Υ.**, που είναι

- (α) Το Τμήμα Εισόδου (Input)
- (β) Το Τμήμα Εξόδου (Output)
- (γ) Το Τμήμα Απομνημονεύσεως (Memory)
- (δ) Το Τμήμα Ελέγχου (Control)
- (ε) Το Τμήμα Αριθμητικής & και Λογικής (Arithmetic and Logic Unit - ALU).

Συνήθως τα τμήματα αριθμητικής, απομνημονεύσεως και ελέγχου θεωρούνται ότι αποτελούν **την κεντρική υπολογιστή μονάδα** (Central Processing Unit - CPU).

Σχήμα 1. Κλασικό Διάγραμμα ψηφιακού Η.Υ.



3. Τρόπος λειτουργίας των Η.Υ.

Η ακριβής εκτέλεση των διαφόρων προγραμμάτων, τα οποία υποβάλλονται (για εκτέλεση) στον Η.Υ., επιτυγχάνεται με τη βοήθεια ενός εποπτεύοντος προγράμματος, μονίμως αποθηκευμένου στον υπολογιστή, **του λειτουργικού συστήματος** (Operating System). Το λειτουργικό σύστημα, εκτός των άλλων, αξιοποιεί και **τις προκατασκευασμένες διαδικασίες** (Software support) που συνοδεύουν κάθε Η.Υ. (Hardware) για τη δραστηριοποίηση των μηχανικών δυνατοτήτων του. Έτσι, λοιπόν, το κάθε πρόγραμμα με **τα δεδομένα του εισάγονται στον Η.Υ. με τη βοήθεια της καταλλήλου μονάδος εισόδου** (Input unit) και την εποπτεία του λειτουργικού συστήματος. Αυτά **αποθηκεύονται** στο ειδικό τμήμα του Η.Υ., **τη μνήμη** (Memory - Store). Οι αριθμητικές και λογικές πράξεις εκτελούνται στο τμήμα της αριθμητικής (Arithmetic - ALU), ενώ **τα αποτελέσματα** της επεξεργασίας **δίδονται** στην επιθυμητή μορφή από τη σχετική **μονάδα εξόδου** (Output unit). Η όλη **επεξεργασία του προγράμματος επιτυγχάνεται με τη βοήθεια του τμήματος ελέγχου** (Control unit) του υπολογιστή.

4. Η έννοια του προγράμματος

Για την επιλύσει ο Η.Υ. ένα συγκεκριμένο πρόβλημα έχει ανάγκη από **μια σειρά εντολών** (οδηγιών), που καθορίζουν τον τρόπο εργασίας του, με κάθε λεπτομέρεια (**αλγόριθμος επίλυσης**) για τη λύση του συγκεκριμένου προβλήματος και, συνήθως, **από ένα σύνολο δεδομένων** (Data), του προβλήματος που θα επεξεργαστεί· ο **συνδυασμός αυτός των εντολών και των δεδομένων καλείται πρόγραμμα**. Φυσικά, το πρόγραμμα προσδιορίζει με κάθε λεπτομέρεια τις απαιτούμενες ενέργειες, για τη λύση του προβλήματος από τον Η.Υ., σε κάθε δυνατή περίπτωση που μπορεί να εμφανιστεί κατά την επεξεργασία των δεδομένων του.

5. Γλώσσα μηχανής - Συμβολική γλώσσα - Διεκπεραιωτές

Δυστυχώς ο Η.Υ. δε «γνωρίζει» την καθομιλουμένη γλώσσα και έτσι **εμείς πρέπει να του μιλήσουμε στη γλώσσα του, τη γλώσσα μηχανής** (Machine language), που βασικά αποτελείται από ένα πεπερασμένο σύνολο από κωδικούς αριθμούς, στους οποίους «ανταποκρίνεται» με μιά ορισμένη λειτουργία (βλέπε πίνακας 1 για το μέγιστο πλήθος των βασικών εντολών στις διάφορες γενιές υπολογιστών). Η πείρα από την αρχική χρήση των γλωσσών μηχανής (Γ.Μ.) ήταν τέτοια, ώστε οι δυσκολίες που ενεφανίσθησαν ανάγκασαν την ανάληψη προσπαθειών **για τη δημιουργία συμβολικών γλωσσών** - (Σ.Γ. - Symbolic languages), περισσότερο εύχρηστων και ευκολοτέρων στην αφομοίωση (κυρίως λόγω χρήσεως συμβολικών ονομάτων και όχι αριθμών) από τις γλώσσες μηχανής. Τούτο έγινε δυνατό όχι διότι οι Η.Υ. «άλλαξαν γλώσσα», αλλά διότι έγινε δυνατή η επινόηση «λεξικών» με τα οποία οι Η.Υ. «μεταφράζουν» οι ίδιοι το συμβολικό πρόγραμμα στη γλώσσα τους.

Τα λεξικά αυτά, που θα τα ονομάζουμε «Διεκπεραιωτές» (Processors, Translators, etc.), είναι ολόκληρα προγράμματα γραμμένα σε Γ.Μ., δέχονται προγράμματα γραμμένα σε Σ.Γ. (**Source programs**) και δημιουργούν αντίστοιχα προγράμματα σε Γ.Μ. (**Object programs**), τα οποία και μόνο δύνανται να εκτελεσθούν από τον Η.Υ.

6. Γλώσσες υψηλού επιπέδου

Οι πρώτες Σ.Γ. ήταν προσανατολισμένες στη φρασεολογία του «ρεπερτορίου» των εντολών των κατανοήσιμων από τον Η.Υ. (Machine oriented languages). Τούτο όμως, λόγω των ειδικών γνώσεων που προϋπέθετε, απέκλειε τη χρήση του Η.Υ. από το ευρύ επιστημονικό και τεχνικό κοινό. Έτσι δημιουργήθηκαν οι **συμβολικές γλώσσες υψηλού επιπέδου** (High level languages) που είναι προσανατολισμένες στις ανάγκες των προβλημάτων (Problem Oriented languages).

Π.χ. η γλώσσα **FORTRAN** (FORmula TRANslator) έγινε με κριτήριο τις ανάγκες **των επιστημονικών προβλημάτων**, η γλώσσα **COBOL** (Common Business Oriented Language) σχεδιάστηκε με κριτήριο τις ανάγκες **των εμπορικών προβλημάτων**, ενώ η γλώσσα **Java** επινοήθηκε με βάση τις ανάγκες του **διαδικτύου**. Το **σπουδαίο με όλες τις γλώσσες υψηλού επιπέδου (Γ.Υ.Ε.), είναι η σχεδόν πλήρης ανεξαρτησία των από τον συγκεκριμένο Η.Υ., και τούτο διότι δεν προϋποθέτουν γνώσεις δομής και λειτουργίας του Η.Υ.**

7. Παράδειγμα χρήσης Γ.Μ., Σ.Γ. και Γ.Υ.Ε.

Το παρακάτω παράδειγμα θα κάνει εναργέστερα τα πλεονεκτήματα των Γ.Υ.Ε. **FORTRAN/Java**, σχετικά με τις Γ.Μ. και τις Σ.Γ. τις προσανατολισμένες στη γλώσσα του Η.Υ.

Ας υποθέσουμε ότι στις παρακάτω θέσεις μνήμης ενός ηλεκτρονικού υπολογιστή υπάρχουν οι αντίστοιχοι αριθμοί (φυσικά στο δυαδικό σύστημα), όπως παρουσιάζονται στον παρακάτω πίνακα 2:

Πίνακας 2. Τμήμα Μνήμης Η.Υ.

Θέση Μνήμης	101010	101011	101100	101101	101110	101111
Αριθμός	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>

Επί πλέον ας υποθέσουμε το ακόλουθο υποσύνολο βασικών εντολών, του πίνακα 3, που αναγνωρίζει ένας συγκεκριμένος υπολογιστής.

Τέλος, εάν υποθέσουμε ότι για την επίλυση ενός προβλήματος ο υπολογισμός της ακόλουθης παράστασης είναι αναγκαίος:

$$F = \frac{A - B + C}{E} \cdot D$$

σε γλώσσα μηχανής, σε συμβολική γλώσσα και στις γλώσσες Υ.Ε. **FORTRAN/Java**, για τους αριθμούς που δίδονται παραπάνω, στον πίνακα 2, τότε τα αντίστοιχα προγράμματα δίδονται στον παρακάτω πίνακα 4.

Πίνακας 3. Τμήμα Εντολών Συμβολικής Γλώσσας

Κωδικός Εντολής	Συμβολικό όνομα	Περιγραφή Ενέργειας Εντολής
100000	LD Δ.Μ.	Τοποθέτηση Περιεχομένου (της) Ορισμένης Διεύθυνσης Μνήμης (ΠΟΔΜ) που ακολουθεί στον α θ ρ ο ι σ τ ή (accumulator)
100001	ADD Δ.Μ.	Πρόσθεση ΠΟΔΜ στο περιεχόμενο του αθροιστή
100010	SUB Δ.Μ.	Αφαίρεση ΠΟΔΜ από το περιεχόμενο του αθροιστή
100011	MUL Δ.Μ.	Πολλαπλασιασμός ΠΟΔΜ με το Περιεχόμενο (Π) του αθροιστή και τοποθέτηση του γινομένου στον αθροιστή
100100	DIV Δ.Μ.	Διαίρεση Π αθροιστή δια ΠΟΔΜ και τοποθέτηση του πηλίκου στον αθροιστή
100101	ST Δ.Μ.	Τοποθέτηση Π αθροιστή στην ΟΔΜ, που ακολουθεί
100111	ENT Δ.Μ.	Ανάγνωση ενός αριθμού και τοποθέτηση αυτού στην ΟΔΜ, που ακολουθεί.

Πίνακας 4. Προγράμματα σε Γ.Μ., σε Σ.Γ. και Γ.Υ.Ε.

ΓΛΩΣΣΑ ΜΗΧΑΝΗΣ	ΣΥΜΒΟΛΙΚΗ ΓΛΩΣΣΑ	(ΜΑΘΗΜΑΤΙΚΟΙ ΥΠΟΛΟΓΙΣΜΟΙ)	ΓΛΩΣΣΕΣ Υ.Ε FORTRAN/Java
100000 101010	<i>LD A</i>	A	
100010 101011	<i>SUB B</i>	$A - B$	
100001 101100	<i>ADD C</i>	$A - B + C$	
100011 101101	<i>MUL D</i>	$(A - B + C) \cdot D$	$F = (A - B + C) * D/E$
100100 101110	<i>DIV E</i>	$(A - B + C) \cdot D/E$	
100101 101111	<i>ST F</i>	$F = (A - B + C) \cdot D/E$	

Σημείωση: Το παραπάνω παραστατικό παράδειγμα δίδει το βασικό χαρακτηριστικό των Γ.Υ.Ε. που είναι η πολλαπλότητα των εργασιών που μπορούν να ενεργοποιηθούν με μια εντολή, σε αντίθεση με τις γλώσσες τις προσανατολισμένες στη μηχανή, όπου κάθε εντολή ενεργοποιεί μόνο μία εργασία.

Τέλος, τα πλεονεκτήματα των Γ.Υ.Ε. είναι ο **εύκολος προγραμματισμός** (συνεπώς και ευρύτερη βάση εφαρμογών από τους ενδιαφερόμενους) και η δυνατότητα συντόμου **εντοπισμού** προγραμματικών **σφαλμάτων**. το μειονέκτημά τους είναι η δύσκολη αριστοποίηση του όλου προγράμματος.

8. Τα τρία βασικά συστήματα μηχανής επεξεργασίας πληροφοριών

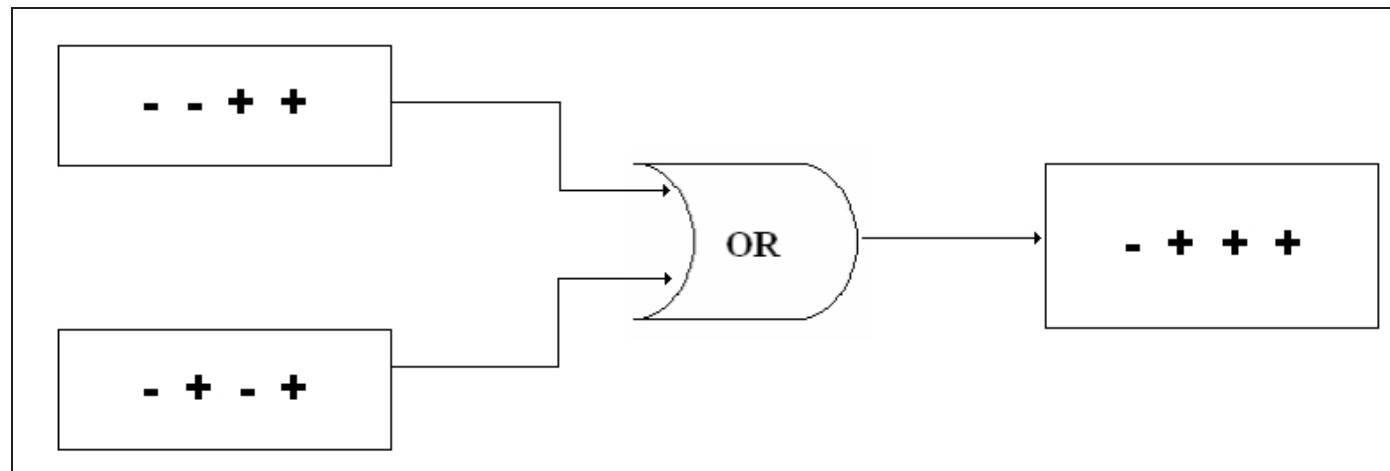
Το κέντρο της «**δυναμικής φιλοσοφίας**» των Η.Υ. είναι η **πραγματικότητα των διακοπών**, που μπορούν να είναι ανοικτοί ή κλειστοί, **των γραμμών επικοινωνίας**, που μεταφέρουν ένα δυαδικό σήμα ή όχι και **των μέσων απομνημόνευσης**, που αποθηκεύουν δυαδικά ψηφία. Επιπρόσθετα δε και **τα τρία βασικά συστήματα - ή πύλες (gates)** - επί των οποίων εδράζεται η δυναμική της ικανότητας επεξεργασίας πληροφοριών των Η.Υ., που είναι διατάξεις που η είσοδος των (input) δέχονται δυαδικές πληροφορίες και παρόμοια η έξοδος (output) των αποδίδει δυαδική πληροφορία.

Πιο συγκεκριμένα έχουμε:

(α) Το σύστημα OR (-ή-)

Το βασικό σύστημα OR αποτελεί την πρώτη βασική διάταξη, με δύο δυαδικές εισόδους και μία έξοδο, ομοίως δυαδική, που εάν εξομοιώσουμε την μία κατάσταση με + (π.χ. ύπαρξη ρεύματος στην αντίστοιχη είσοδο) και την άλλη με - (απουσία ρεύματος π.χ.), τότε η λειτουργία της αποδίδεται από την παρακάτω σχηματική διάταξη 2, που περιγράφει τον τρόπο λειτουργίας της σε κάθε δυνατή περίπτωση, των δεδομένων της εισόδου της.

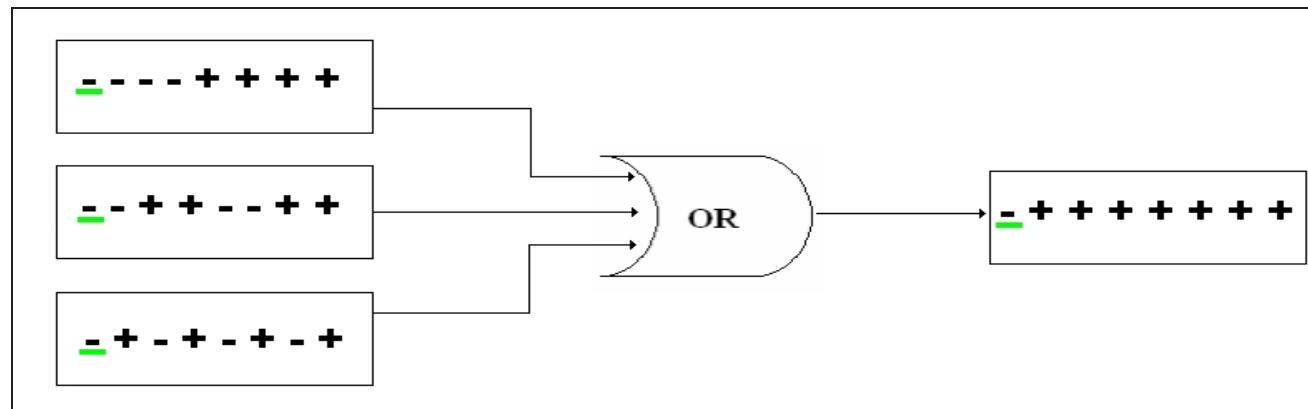
Σχήμα 2. Η διάταξη OR



(Η σειρά των καταστάσεων στις εισόδους και στην έξοδο αποδίδουν την πραγματική λειτουργία). Το σημαντικό στη λειτουργία του συστήματος είναι η ιδιότητα ότι αποδίδει (+) όταν **τουλάχιστον μία** είσοδος είναι στην κατάσταση (+). Ή ισοδύναμα, αποδίδει την κατάσταση (-), όταν **όλες οι εισόδους** του είναι στην κατάσταση (-).

Το ενδιαφέρον είναι ότι τα παραπάνω χαρακτηριστικά βοηθούν στη γενίκευση του συστήματος με περισσότερες από 2 εισόδους· π.χ. για την περίπτωση του OR με 3 εισόδους θα έχουμε την διάταξη του σχήματος 3:

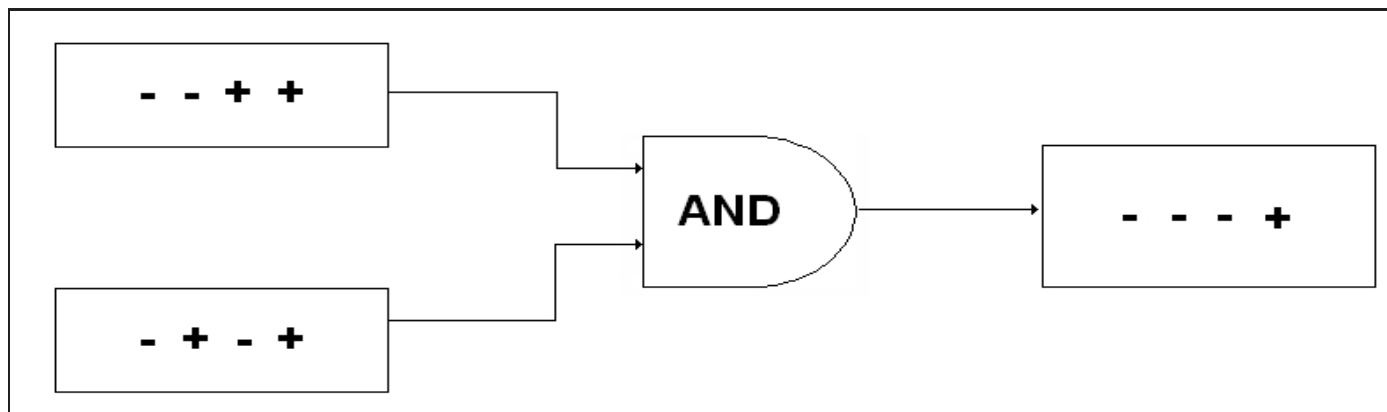
Σχήμα 3. Η διάταξη OR με πολλαπλές εισόδους (3 στο παράδειγμα)



(β) Το σύστημα AND (-και-)

Το βασικό σύστημα AND αποτελεί τη δεύτερη βασική διάταξη, με δύο δυαδικές εισόδους και μία έξοδο, όμοια δυαδική, που η λειτουργία της αποδίδεται από την ακόλουθη σχηματική διάταξη 4 (που περιγράφει τον τρόπο λειτουργίας της σε κάθε δυνατή περίπτωση):

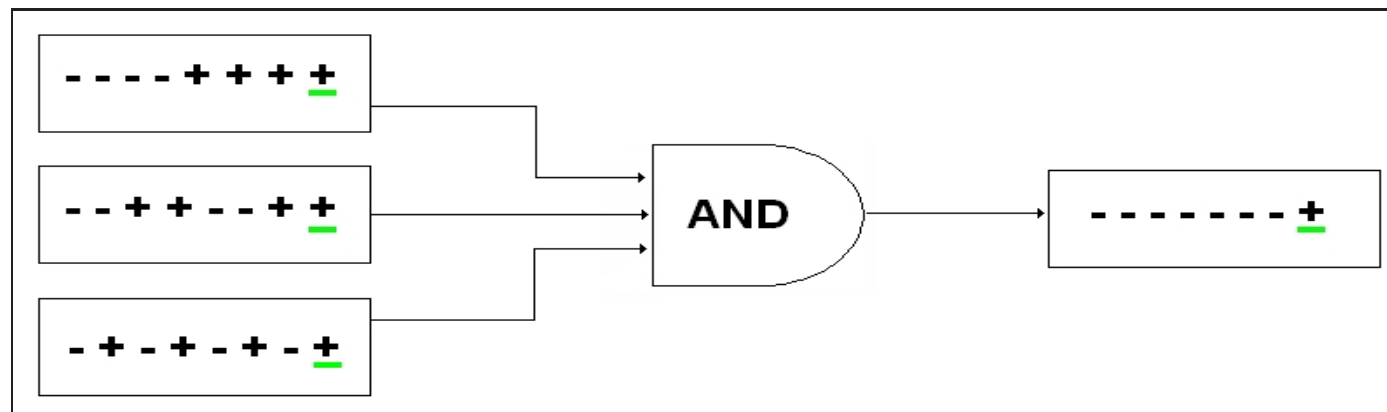
Σχήμα 4. Η διάταξη AND



Το σημαντικό στη λειτουργία του συστήματος είναι η ιδιότητα ότι αποδίδει την κατάσταση (+) όταν **όλες οι εισοδοί του** είναι στην κατάσταση (+): σε κάθε άλλη περίπτωση αποδίδει την κατάσταση (-).

Το ανωτέρω χαρακτηριστικό του συστήματος βοηθά στην επέκταση (γενίκευση) του σε πολλαπλές εισόδους· π.χ. στην περίπτωση των 3 εισόδων θα έχουμε τη διάταξη, με την ενδεικνυόμενη λειτουργία, του σχήματος 5:

Σχήμα 5. Η διάταξη AND με πολλαπλές εισόδους (3 στο παράδειγμα)



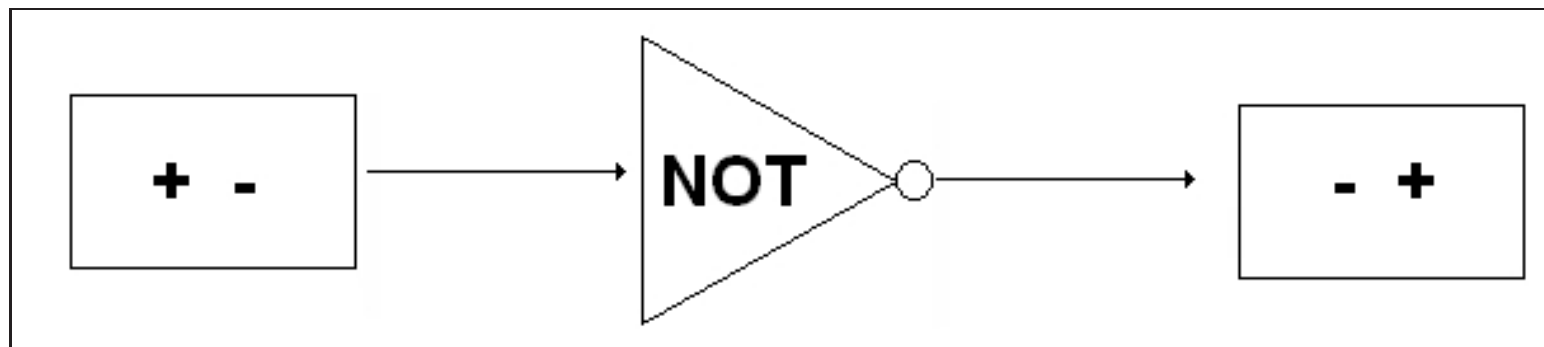
(γ) Το σύστημα NOT (-όχι-)

Το **τρίτο** και τελευταίο **βασικό σύστημα είναι το NOT**, που αποτελείται από μία διάταξη με μία και μόνο είσοδο, και μία έξοδο, η λειτουργία της οποίας είναι απλούστατη και αποδίδεται από **την αντιστροφή της καταστάσεως εισόδου**.

Όταν η είσοδος είναι σε κατάσταση (+), τότε η έξοδος αποδίδει (-), ενώ όταν η είσοδος είναι σε κατάσταση (-) η έξοδος αποδίδει την κατάσταση (+).

Το παρακάτω σχήμα 6 αποδίδει τον συμβολισμό της διάταξης και τον τρόπο λειτουργίας της:

Σχήμα 6. Η διάταξη NOT



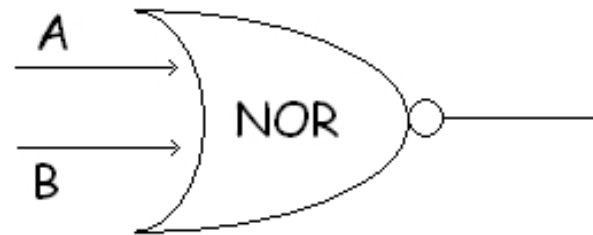
Παρατηρήσεις:

α. Η ονομασία των παραπάνω τριών Σ.Μ.Ε.Π. ως «πυλών» καθίσταται αυτονόητη αν λάβουμε μία διάταξη AND και θέσουμε **μία της είσοδο σε κατάσταση +**· τότε από τον τρόπο λειτουργίας της, η έξοδό της θα είναι **η κατάσταση της άλλης εισόδου**, δηλαδή η AND μεταβιβάζει την κατάσταση της δεύτερης εισόδου που απλώς διέρχεται από την πύλη. Αντίθετα, αν **η μία είσοδος είναι σε κατάσταση -**· τότε, η έξοδος της NAND θα είναι σε κατάσταση -, ανεξαρτήτως της κατάστασης της δεύτερης εισόδου· δηλαδή **η AND λειτουργεί ως φραγμός-διακόπτης για τη δεύτερη είσοδο**.

β. Είναι ιδιαίτερα βοηθητικοί οι συνδυασμοί των δύο πρώτων πυλών, OR και AND με την πύλη OR. Τα δε ζεύγη πυλών που προκύπτουν φέρουν την ονομασία NOR και NAND αντίστοιχα, οι δε πίνακες λειτουργίας των καθώς και ο συμβολισμός τους αποδίδονται από τα παρακάτω σχήματα 6.1 και 6.2.

Σχήμα 6.1. Η πύλη (NOR)

A	B	NOR
+	+	-
+	-	-
-	+	-
-	-	+



Σχήμα 6.2. Η πύλη (NAND)

A	B	NAND
+	+	-
+	-	+
-	+	+
-	-	+



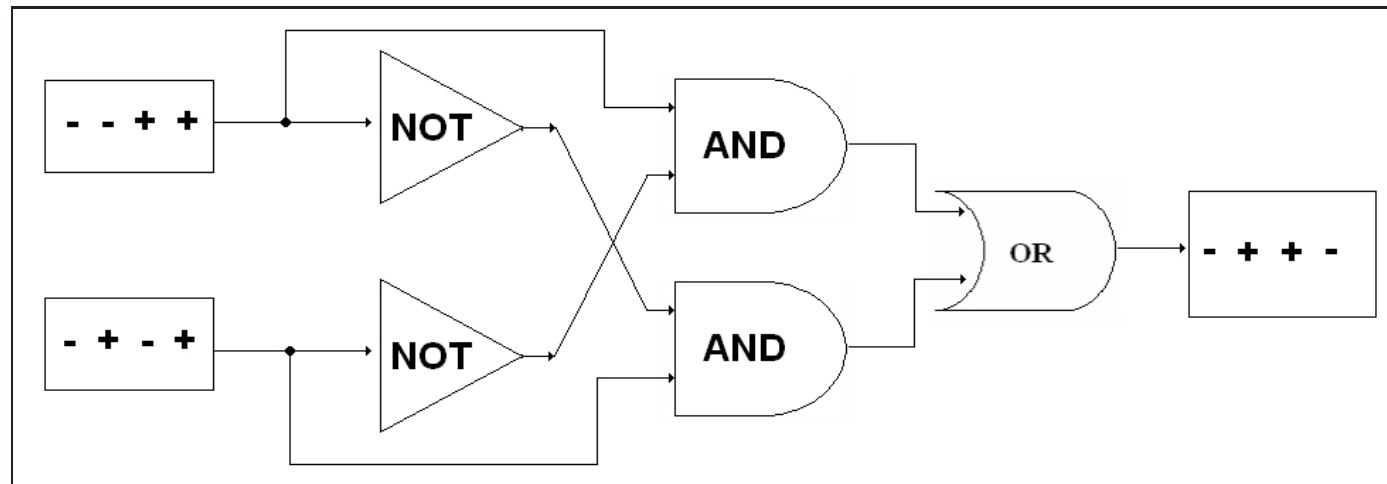
γ. Οι παραπάνω δύο πύλες NOR και NAND έχουν την ιδιότητα με κατάλληλους συνδυασμούς να λειτουργούν η κάθε μία τους ως OR, AND και NOT, πράγμα που τις καθιστά **οικουμενικές** (universal). Π.χ. εάν οι δύο εισοδοι είναι ίδιες τότε λειτουργούν ως πύλες NOT (να διαπιστωθεί από τον πίνακα λειτουργίας ότι η έξοδος τους είναι το συμπλήρωμα της εισόδου). Αυτή η οικουμενικότητα κάνει τη συγκεκριμένη πύλη NAND να είναι το μοναδικό δομικό υλικό που κατασκευάζονται οι σύγχρονοι Η.Υ.

9. Εφαρμογές

(α) Ο Συγκριτής (Comparator)

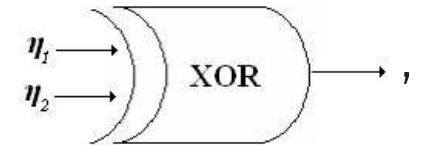
Μια ενδιαφέρουσα και εξόχως βοηθητική διάταξη, που είναι εφαρμογή των τριών παραπάνω βασικών Σ.Μ.Ε.Π., είναι ο **συγκριτής**, που μας δίνει τη δυνατότητα να διακρίνουμε εάν δύο δυαδικές πληροφορίες είναι της ίδιας καταστάσεως, ή διαφορετικές. Η δομή και η λειτουργία του αποδίδεται από το παρακάτω σχήμα 7:

Σχήμα 7. Η διάταξη του Συγκριτή



Το ενδιαφέρον από το παραπάνω σχήμα είναι το γεγονός, ότι ο συγκριτής αποδίδει κατάσταση (+) όταν οι δυαδικές εισοδοί είναι διαφορετικές, ενώ αποδίδει κατάσταση (-) όταν οι εισοδοί είναι της ίδιας καταστάσεως.

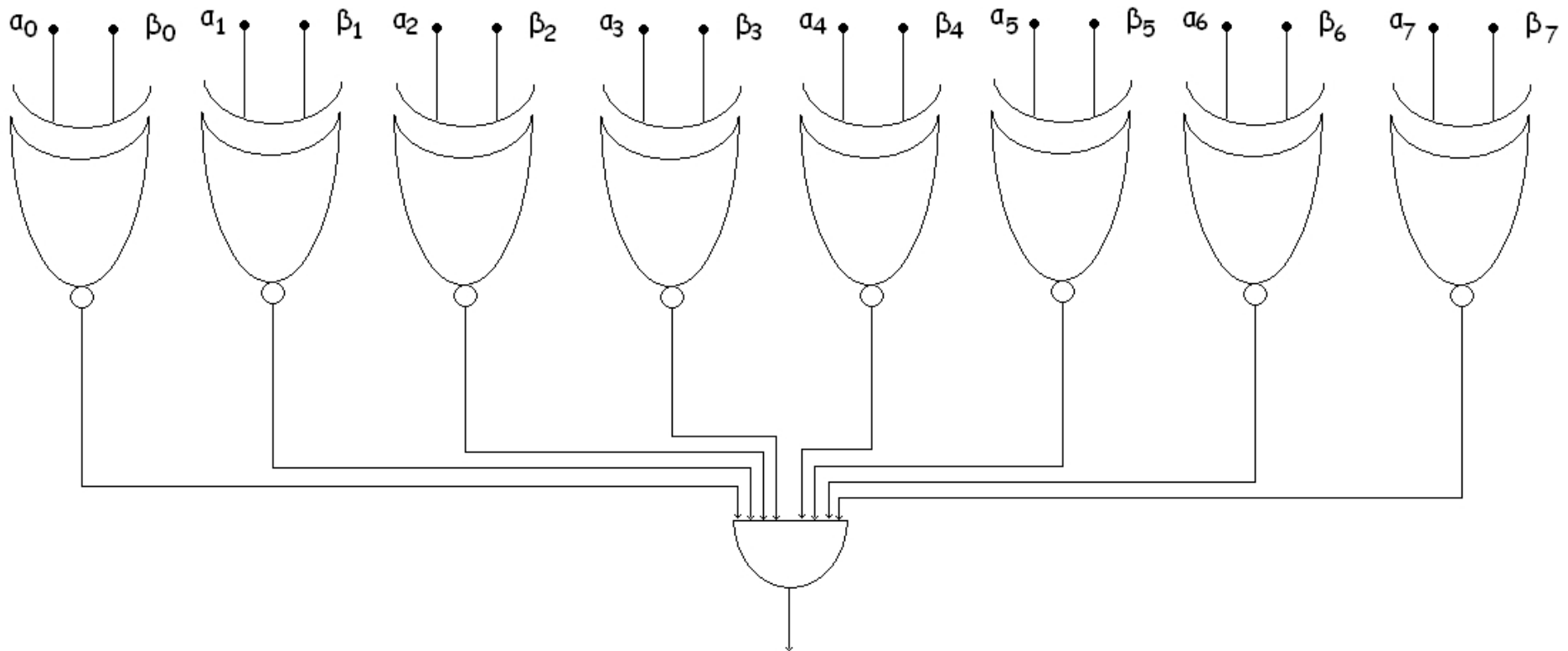
(Σημείωση: Η σημασία του συγκριτή είναι τόσο μεγάλη ώστε η όλη διάταξη να αποτελεί την 4η βασική πύλη, με την ονομασία «XOR», που αποδίδεται με το σύμβολο



ο δε συνδυασμός της με τη πύλη NOT δημιουργεί την πύλη XNOR με την χαρακτηριστική ιδιότητα να αποδίδει κατάσταση + όταν οι 2 εισοδοί είναι της ίδιας καταστάσεως και κατάσταση -, όταν οι καταστάσεις της εισόδου είναι διαφορετικές, πράγμα που αξιοποιείται για τη δημιουργία του **συγκριτή λέξεων** (word comparator). Ως γνωστό, στην μνήμη του υπολογιστή οι πληροφορίες αποθηκεύονται σε λέξεις, που είναι πολλαπλάσια του 1 byte.

Εάν υποθέσουμε ότι έχουμε να συγκρίνουμε τα περιεχόμενα δύο λέξεων (μήκους 1 byte) $a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7$ και $b_0 b_1 b_2 b_3 b_4 b_5 b_6 b_7$ της μνήμης, τότε η ακόλουθη διάταξη αποδίδει κατάσταση + εάν οι δύο λέξεις είναι ίδιες και κατάσταση - εάν διαφέρουν.

Σχήμα 7.1. Συγκριτής 2 λέξεων μήκους 1 (byte)



(β) Εφαρμογή στην άλγεβρα BOOLE

Μια εξόχως σημαντική εφαρμογή των διατάξεων OR, AND και NOT βρίσκεται στην **Άλγεβρα του Boole** που πρώτα αξιοποιήθηκε **στη λογική των προτάσεων**. Ο G. Boole ανέπτυξε την ομώνυμη Άλγεβρα σε δύο έργα του: «**Η Μαθηματική Ανάλυση της Λογικής**»* και «**Μια Διερεύνηση του Νόμων της Σκέψης**»† με στόχο να ανακαλύψει τους τρόπους εργασίας του ανθρώπινου μυαλού. Το ενδιαφέρον στην προκειμένη περίπτωση είναι ότι **η Άλγεβρα του Boole μπορεί εξίσου να εφαρμόζεται και στο σχεδιασμό κυκλωμάτων Η.Υ.** Μια άλγεβρα Boole είναι **η άλγεβρα της Λογικής ή η Άλγεβρα των υποσυνόλων ενός συνόλου.**

*«The Mathematical Analysis of Logic», 1847

†«An Investigation of the Laws of Thoughts», 1854

Θα ήταν βοηθητικό στη συνέχεια, τις δύο καταστάσεις + και - που χρησιμοποιήσαμε για την παρουσία ή μη ρεύματος σ' ένα αγωγό να τα αντικαταστήσουμε με τις 2 τιμές 1 και 0 που μπορούν να πάρουν οι μεταβλητές BOOLE*. Έτσι θα μπορέσουμε ευκολονόητα να δώσουμε τις πράξεις της πρόσθεσης (+), του πολλαπλασιασμού (·) και της αντιστροφής (-) των μεταβλητών Boole με τη βοήθεια των διατάξεων OR, AND και NOT.

(i). Πρόσθεση Boole

Πράγματι, δύο μεταβλητές Boole A και B μπορούν να προστεθούν και το άθροισμά της γράφεται $A+B$, με τιμές που δίδονται από τον παρακάτω πίνακα αληθείας:

Σχήμα 8. Πίνακας αληθείας της πρόσθεσης Boole

	A	0	1
B			
0		0	1
1		1	1

*Στους Η.Υ. ως δίτιμα στοιχεία χρησιμοποιούνται τα flip-flops ή μαγνητικοί δακτύλιοι κ.λ.π.

(Παρατηρήσατε ότι οι τιμές του αθροίσματος είναι ακριβώς οι ίδιες που προκύπτουν για την διάταξη OR - βλέπε σχήμα 2 - που κατά συνέπεια υλοποιεί τον τελεστή της πρόσθεσης Boole).

(ii). Πολλαπλασιασμός Boole

Παρόμοια, ο πολλαπλασιασμός (\cdot) δύο μεταβλητών Boole A και B αποδίδεται από την παράσταση $A \cdot B$ και έχει τον παρακάτω πίνακα αληθείας:

Σχήμα 9. Πίνακας αληθείας του πολλαπλασιασμού Boole

	A	
B	0	1
0	0	0
1	0	1

(Παρατηρήσατε ότι το γινόμενο Boole έχει ακριβώς τις ίδιες τιμές που παρουσιάζονται στη διάταξη AND - βλέπε σχήμα 4 -, που κατά συνέπεια υλοποιεί τον τελεστή του πολλαπλασιασμού Boole).

(iii). Αντιστροφή

Η πράξη της αντιστροφής ($-$) είναι η τρίτη και τελευταία πράξη της Άλγεβρας Boole και για την οποιαδήποτε μεταβλητή Boole A έχει τον ακόλουθο πίνακα αληθείας:

Σχήμα 10. Πίνακας αληθείας της αντιστροφής Boole

A	0	1
\bar{A}	1	0

(Παρατηρήσατε ότι ο \bar{A} έχει ακριβώς τις ίδιες τιμές που παρέχει η διάταξη NOT, που κατά συνέπεια υλοποιεί τον τελεστή της αντιστροφής).

Φυσικά, αν οι μεταβλητές Boole λαμβάνουν μια ακολουθία τιμών τότε μπορούμε να εφαρμόσουμε τους παραπάνω τελεστές σε ζεύγη τιμών τους και να λάβουμε μιά ακολουθία αποτελεσμάτων.

Π.χ. εάν είχαμε τις ακολουθίες τιμών:

$$A = 10111001 \text{ και } B = 11000001,$$

τότε θα είχαμε(για το άθροισμα, γινόμενο και αντίστροφο A):

$$A + B = 11111001, \quad A \cdot B = 10000001, \quad \bar{A} = 01000110$$

(Προφανώς το σύμβολο της ισότητας παραπάνω έχει την έννοια: «είναι ακριβώς η ίδια ακολουθία ως η»).

(γ) Δημιουργία μηχανών ειδικού σκοπού

Στην εφαρμογή αυτή θα δούμε πως μπορούμε να επινοήσουμε μηχανές για την επίλυση λογικών προβλημάτων. Προτού όμως συνεχίσουμε θα ήταν βοηθητικό να τονίσουμε **τα τρία βασικά θέματα που ξεκαθάρισε ο Boole για τις εφαρμογές της Άλγεβρας του, στη λογική των προτάσεων.**

Θέμα 1ο: Μια πρόταση μπορεί να είναι είτε μόνο αληθής είτε μόνο ψευδής· ποτέ δεν μπορεί να είναι ταυτόχρονα και αληθής και ψευδής.

Θέμα 2ο Μπορούμε να δημιουργήσουμε σύνθετες προτάσεις με χρήση των τριών συνθετικών **ή** (or), **και** (and), **δεν** (not).

Θέμα 3ο Μπορούμε να προσδιορίσουμε το αληθές ή το ψευδές μιάς πρότασης, μηχανικά.

Τα παρακάτω παραδείγματα θα κάνουν σαφέστερες τις έννοιες.

Παράδειγμα 1ο: Σ' ένα σπιτικό με δύο παιδιά (αγόρι και κορίτσι) μπορούμε να διατυπώσουμε τις παρακάτω προτάσεις για τις 7 ημέρες της εβδομάδας, που αποδίδονται στον παρακάτω πίνακα 5:

	Πίνακας 5.						
Προτάσεις	Δ	Τ	Τ	Π	Π	Σ	Κ
A: Η μητέρα μου είναι άρρωστη	0	1	1	0	0	0	1
B: Ο πατέρας μου είναι άρρωστος	1	1	1	0	1	0	1
Γ: Ο αδελφός μου είναι άρρωστος	0	0	1	0	1	0	0
Δ: Η αδελφή μου είναι άρρωστη	0	0	1	0	0	0	1

Φυσικά, ανάλογα με την θεραπευτική αγωγή που ακολουθείται είναι δυνατό να συμβεί σε ορισμένες μέρες να είναι άρρωστοι ορισμένα μέλη του σπιτικού και σε άλλες ημέρες να μην είναι. Το γεγονός της αρρώστιας καθιστά τις προτάσεις αληθείς, οπότε για τις 7 ημέρες της εβδομάδας μπορούμε να έχουμε τον προηγούμενο πίνακα αληθείας (Πίνακας 5).

Στη συνέχεια μπορούμε να δημιουργήσουμε σύνθετες προτάσεις της μορφής:

Πρόταση (1): «Η αδελφή μου είναι άρρωστη ή ο αδελφός μου είναι άρρωστος»,

δηλαδή, οι προτάσεις Γ και Δ έχουν συνδεθεί με το «ή», που για μας θα σημαίνει: «κάποιος από τους δύο ή αμφότεροι».

Ο πίνακας αληθείας της πρότασης (1) για τις ημέρες της εβδομάδας που ενδιαφέρει είναι: 0 0 1 0 1 0 1, δηλαδή συμπίπτει με τον πίνακα αληθείας του αθροίσματος Boole των προτάσεων Γ και Δ . Κατά συνέπεια μπορούμε να το συμβολίσουμε με:

$$\Gamma + \Delta: 0 0 1 0 1 0 1$$

και να το υπολογίσουμε σαν άθροισμα Boole των προτάσεων Γ και Δ .

Μιά άλλη σύνθετη πρόταση μπορεί να δημιουργηθεί με το συνθετικό «και». Έτσι, έχουμε την πρόταση:

Πρόταση (2): «Η αδελφή μου είναι άρρωστη και ο αδελφός μου είναι άρρωστος».

Προφανώς ο πίνακας αληθείας της πρότασης (2) είναι: 0 0 1
0 0 0 0, και συμπίπτει με τον πίνακα αληθείας του γινομένου Boole των προτάσεων Γ και Δ. Κατά συνέπεια μπορούμε να το συμβολίσουμε με:

$$\Gamma \cdot \Delta : 0 0 1 0 0 0 0,$$

και να το υπολογίσουμε σαν γινόμενο Boole των προτάσεων Γ και Δ.

Τέλος μιά τρίτη σύνθετη πρόταση μπορεί να δημιουργηθεί με χρήση του συνδετικού «δεν». Πιο συγκεκριμένα έχουμε:

Πρόταση (3): «Ο αδελφός μου δεν είναι άρρωστος».

Ο πίνακας αληθεία της (3) είναι: 1 1 0 1 0 1 1, και συμπίπτει με τον πίνακα αληθείας της αντίστροφης πρότασης $\bar{\Gamma}$. κατά συνέπεια μπορούμε να συμβολίσουμε την (3) με:

$$\bar{\Gamma} : 1 1 0 1 0 1 1$$

και να την υπολογίσουμε σαν αντίστροφη πρόταση Boole της Γ .

Από το προηγούμενο παράδειγμα είναι σαφές ότι:

(i) Στις σύνθετες προτάσεις που δημιουργούνται με τα συνθετικά ή, και, δεν το αληθές ή το ψευδές αυτών μπορεί να υπολογιστεί με πράξεις Boole αντί της καθιερωμένης λογικής. Δηλαδή, στην προκειμένη περίπτωση, οι πράξεις Boole είναι πράξεις της λογικής.

(ii) Για την υλοποίηση των πράξεων Boole μπορούμε να αξιοποιήσουμε τις διατάξεις OR, AND και NOT για την μηχανική εκτέλεση των υπολογισμών.

(iii) Θεωρήματα της Άλγεβρας Boole μπορούν να χρησιμοποιηθούν για την απλοποίηση των σύνθετων προτάσεων, πράγμα που θα εξετάσουμε αργότερα με λεπτομέρεια.

Παράδειγμα 2ο: Με τις προτάσεις A , B , Γ και Δ του πίνακα 5 του προηγούμενου παραδείγματος, δημιουργούμε τις παρακάτω σύνθετες προτάσεις E , Z και H , που στη συνέχεια θα μετατρέψουμε σε εκφράσεις Boole και τελικά θα δώσουμε σχετικές μηχανικές διατάξεις για την υλοποίησή τους. Έτσι, έχουμε:

E: Η αδελφή μου και ο αδελφός μου αμφότεροι είναι άρρωστοι.

Z: Η αδελφή μου είναι άρρωστη, αλλά ο αδελφός μου δεν είναι άρρωστος,

H: Ούτε η αδελφή μου, ούτε ο αδελφός μου είναι άρρωστοι.

Εύκολα βλέπουμε ότι οι αντίστοιχες εκφράσεις Boole των E , Z και H είναι:

$$E : \Gamma \cdot \Delta$$

$$Z : \Gamma \cdot \overline{\Delta}$$

$$H : \overline{\Gamma} \cdot \overline{\Delta}.$$

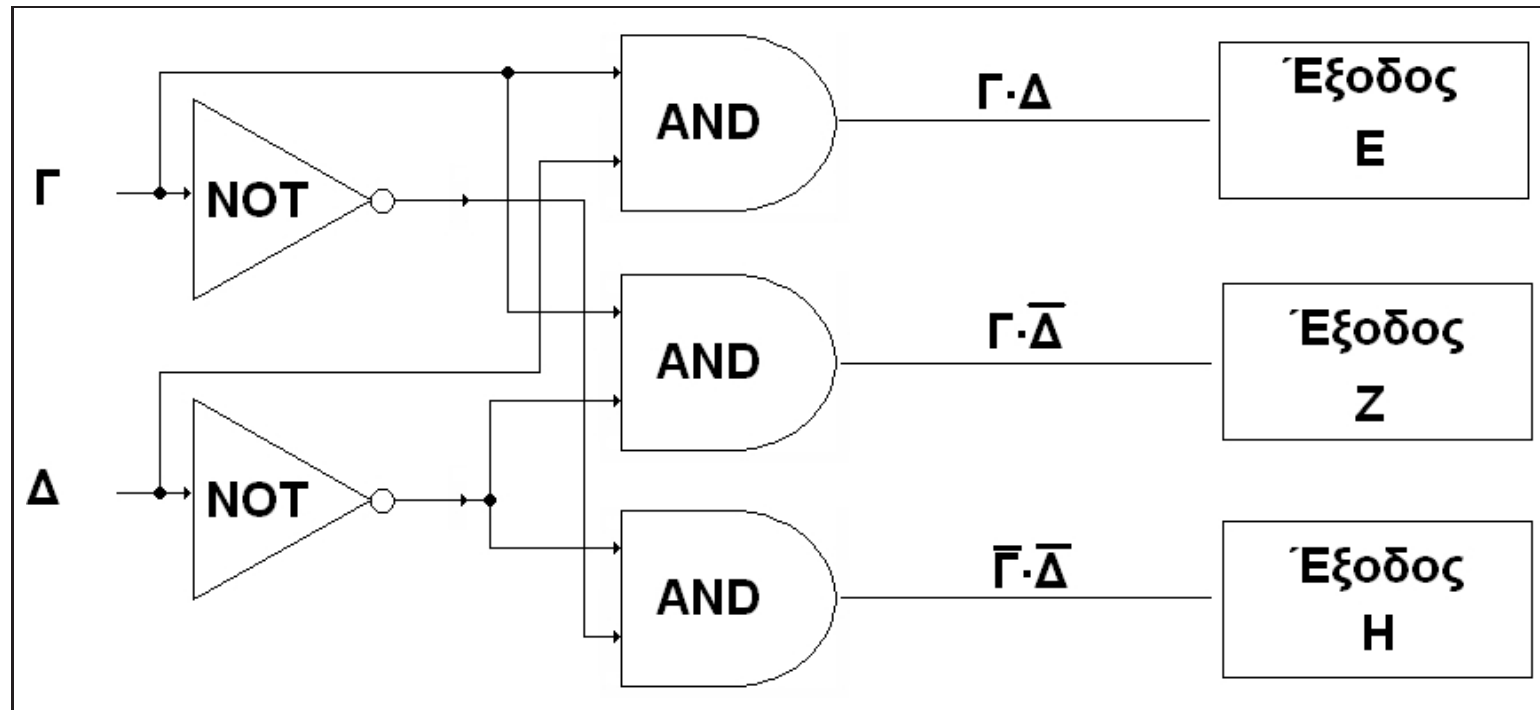
Επιπρόσθετα, ο πίνακας αληθείας των δίδεται από τον ακόλουθο πίνακα 6.

Πίνακας 6.

Γ :	0	0	1	0	1	0	0
Δ :	0	0	1	0	0	0	1
$\bar{\Gamma}$:	1	1	0	1	0	1	1
$\bar{\Delta}$:	1	1	0	1	1	1	0
$\Gamma \cdot \Delta$:	0	0	1	0	0	0	0
$\Gamma \cdot \bar{\Delta}$:	0	0	0	0	1	0	0
$\bar{\Gamma} \cdot \bar{\Delta}$:	1	1	0	1	0	1	0

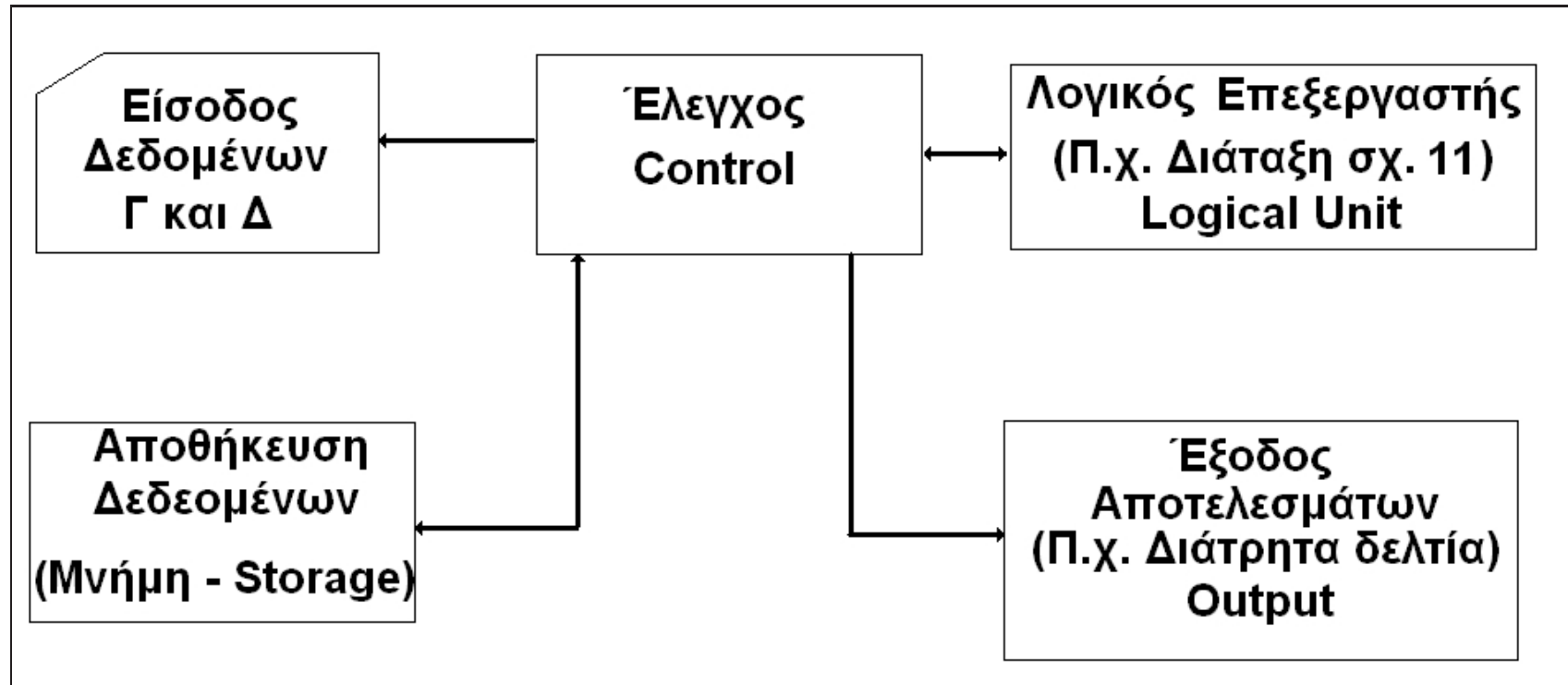
Τέλος, η παρακάτω διάταξη είναι εύκολο να δει κανείς ότι υλοποιεί τις προηγούμενες εκφράσεις Boole: κατά συνέπεια οι έξοδοι E, Z και H θα δίδουν μηχανικά τα επιθυμητά αποτελέσματα, στο παρακάτω σχήμα 11.

Σχήμα 11. Διάταξη λογικής επεξεργασίας



Τέλος, εάν θέλαμε για την παραπάνω επεξεργασία να είχαμε ένα ολοκληρωμένο Σ.Μ.Ε.Π. με είσοδο δεδομένων, δυνατότητα αποθήκευσης και έξοδο αποτελεσμάτων, καθώς και διάταξη ελέγχου (π.χ. για τον έλεγχο του χρονισμού των διαφόρων ενεργειών) θα μπορούσε η παρακάτω διάταξη (**Μηχανή ειδικού σκοπού**) να αποτελούσε ένα τέτοιο σύστημα:

Σχήμα 12. Μηχανή ειδικού σκοπού



(δ) Ο Κωδικοποιητής (Encoder)

Ο πιο άμεσος τρόπος εισαγωγής πληροφοριών στον Η.Υ. είναι το γνωστό μας **πληκτρολόγιο**, του σχήματος 13, που ακολουθεί:

Σχήμα 13. Το πληκτρολόγιο



(παρόμοιο του πληκτρολογίου της γραφομηχανής), με **τις εκατό περίπου διαφορετικές επιλογές στοιχείων-λειτουργιών που μπορεί κανείς να δημιουργήσει**, κάνοντας χρήση των διαφόρων συνδυασμών των πλήκτρων, που περιέχει.

Πάνω στα πλήκτρα του πληκτρολογίου υπάρχουν είτε τα αλφαριθμητικά σύμβολα, ή σύμβολα στίξεως, ή πράξεων που δημιουργεί σε **κωδικοποιημένους δυαδικούς αριθμούς, βάσει ενός κώδικα**, είτε οι εργασίες που μπορεί να ενεργοποιήσεις (λειτουργικά πλήκτρα). Το σπουδαίο σε κάθε περίπτωση είναι η δημιουργία ενός κωδικοποιημένου σήματος σε δυαδική μορφή μήκους 8 δυαδικών ψηφίων (1 Byte) που προωθείται στον Η.Υ. Έτσι, π.χ. εάν υποθέσουμε την **φυσική κωδικοποίηση** για την περίπτωση των δέκα αριθμητικών συμβόλων του δεκαδικού συστήματος, που αποδίδεται από τον παρακάτω πίνακα 7:

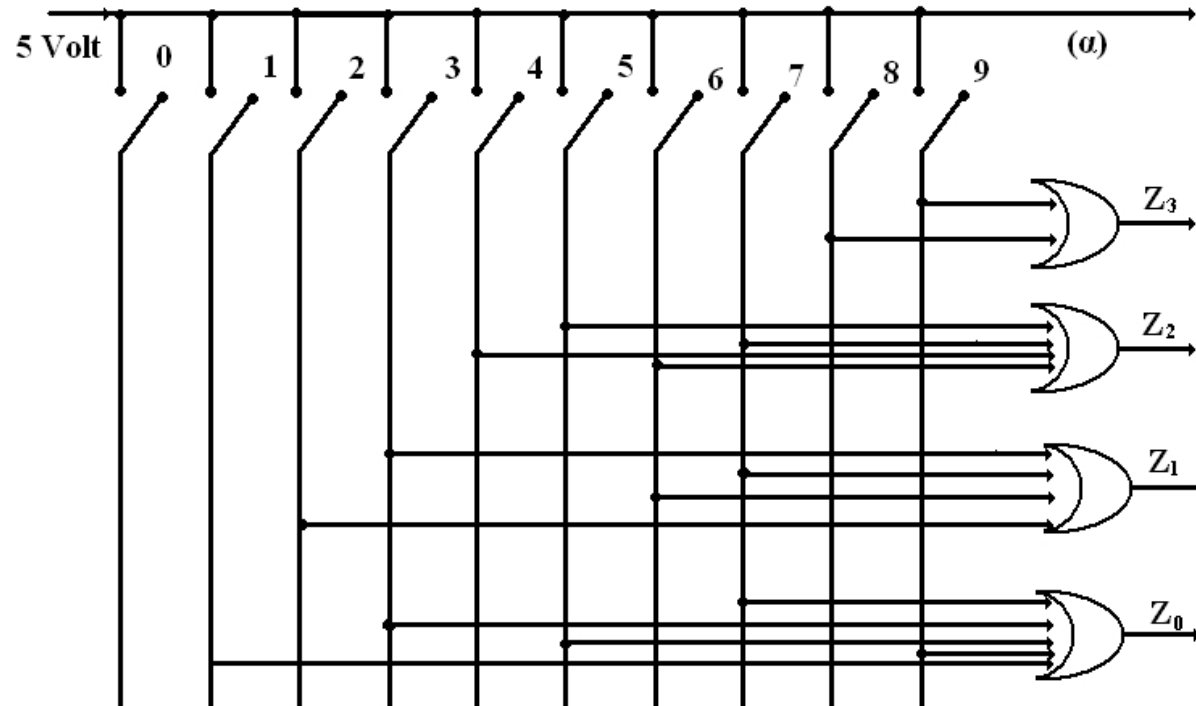
Πίνακας 7: Η φυσική κωδικοποίηση των ψηφίων 0-9

Δεκαδικά ψηφία	Δυαδικοί αριθμοί στο σύστημα 8-4-2-1
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

ο κωδικοποιημένος δυαδικός, που δημιουργείται, έχει τα 4 αρχικά του δυαδικά ψηφία **μηδενικά** και τα επόμενα 4 όπως περιγράφεται στον παραπάνω πίνακα: δηλαδή ο **1** αποδίδεται από τον δυαδικό 00000001 ενώ ο **9** από τον 00001001. Τα 4 πρώτα ψηφία αξιοποιούνται για την κωδικοποίηση των μη αριθμητικών πληροφοριών (π.χ. γραμμάτων ή άλλων συμβόλων).

Μιά διάταξη δημιουργίας των κατάλληλων **αριθμητικών ψηφίων** είναι το παρακάτω σχήμα 14:

Σχήμα 14. Ένας κωδικοποιητής των ψηφίων του Δεκαδικού Συστήματος



Ο $Z_3Z_2Z_1Z_0$ θα είναι ο δυαδικός που παρίσταται στην έξοδο του κωδικοποιητή.

που αποτελεί ένα **κωδικοποιητή** των 10 ψηφίων του δεκαδικού συστήματος κωδικοποιημένου στη φυσική κωδικοποίηση «8-4-2-1».

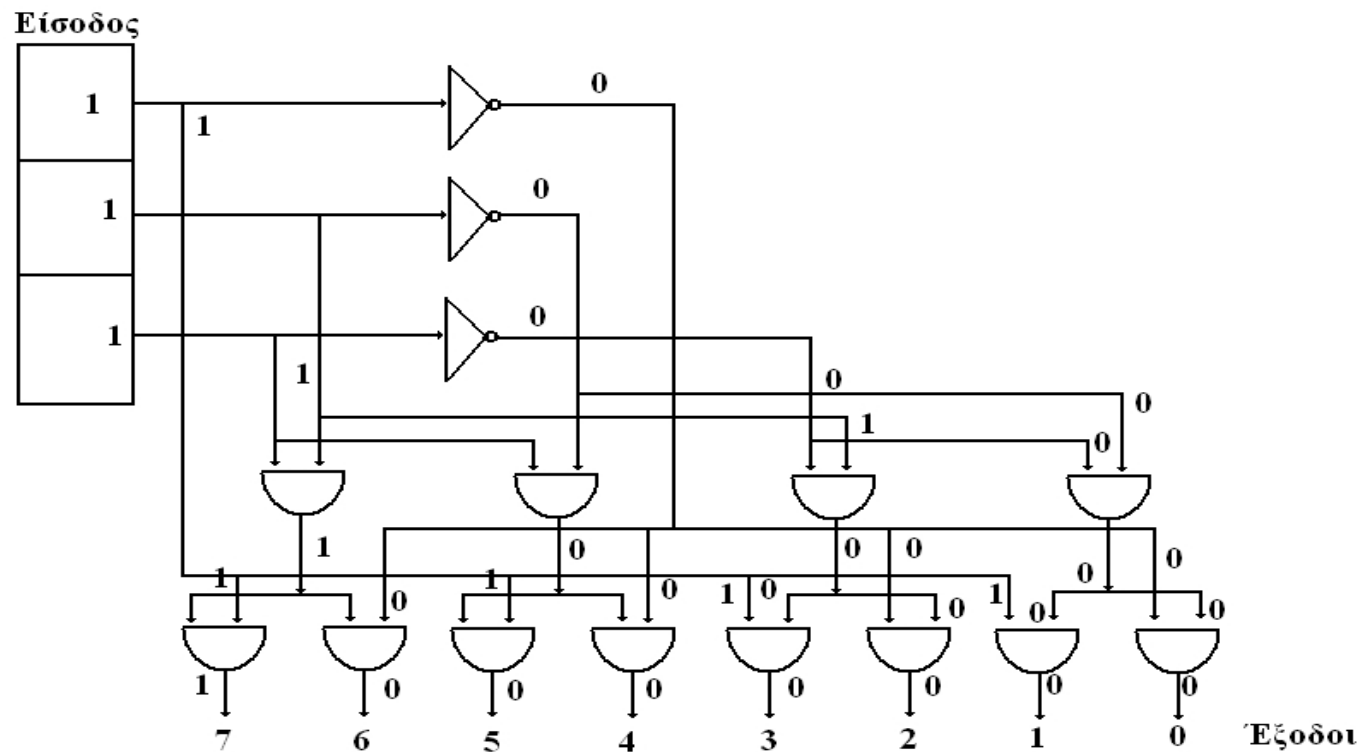
Στο σχήμα 14 είναι προφανές ότι στο πάτημα ενός αριθμητικού πλήκτρου, από τα 10, καθοδηγεί το διαρρέον ρεύμα του αγωγού (α) στις κατάλληλες εξόδους για την δημιουργία των αντίστοιχων δυαδικών καταστάσεων, που αντιστοιχούν στα επιθυμητά ψηφία. Π.χ. το πάτημα του πλήκτρου 5, στέλνει σε μία των εισόδων των πυλών OR, αυτή που συνδέεται με τον κατακόρυφο αγωγό του 5, το ρεύμα που καθιστά, κατά συνέπεια τις εξόδους των πυλών Z_0 και Z_2 ενεργοποιημένες (δηλαδή 1), ενώ οι άλλες θα είναι 0· έτσι ο αριθμός θα είναι ο 0101(=5).

(ε) Ο Αποκωδικοποιητής (Decoder)

Ο αποκωδικοποιητής επιτελεί την αντίστροφη λειτουργία του κωδικοποιητή· δηλαδή **δέχεται ένα δυαδικό αριθμό** και αποδίδει τον αντίστοιχο **δεκαδικό**. Η κύρια εφαρμογή του αποκωδικοποιητή είναι να επιλέγει εκάστοτε την θέση μνήμης που εμπλέκεται στην εκτέλεση μιάς εντολής με βάση την διεύθυνση της, που είναι πάντοτε σε δυαδική μορφή (βλέπε σχετικά στο σχήμα 1 του κλασσικού διαγράμματος ενός ψηφιακού Η.Υ.).

Ένα τριψήφιο αποκωδικοποιητή, που περιέχει τις $8(2^3)$ εξόδους (0-7), αποδίδει το σχήμα 15, που η δομή του είναι τέτοια που ανάλογα με το περιεχόμενο της εισόδου (στην εμφανιζόμενη περίπτωση 111) η αντίστοιχη έξοδος 7, παρέχει με την κατάσταση 1 την ενεργοποίηση που επιθυμούμε, ενώ όλες οι άλλες εξοδοι ευρίσκονται στην κατάσταση μηδέν.

Σχήμα 15. Αποκωδικοποιητής 3 ψηφίων



```
PROGRAM QUADRATIC_EQUATIONS_REALROOTS
DO I = 1, 5
    READ *, A, B, C
    D = B*B - 4*A*C
    R1 = (-B - D**0.5)/(2*A)
    R2 = (-B + D**0.5)/(2*A)
PRINT *, A, B, C, R1, R2
ENDDO
END PROGRAM
```



```

PROGRAM QUADRABEST
READ *, N
DO I = 1, N
  READ *, A, B, C
  IF (A /= 0) THEN          D = B*B - 4*A*C
    R1 = (-B - D**0.5)/(2*A)
    R2 = (-B + D**0.5)/(2*A)
    PRINT *, "REAL ROOTS", R1, R2
  ELSE IF (B /= 0) THEN
    R = - C/B
    PRINT *, "LINEAR EQUATION WITH THE ROOT R=", R
  ELSE IF (C /= 0) THEN
    PRINT *, "THERE IS NO EQUATION, AND, OF COURSE, THERE IS NO ROOT"
  ELSE
    PRINT *, "THE GIVEN EQUATION IS AN IDENTITY", A, B, C
  ENDIF
ENDDO
END PROGRAM QUADRABEST

```